



# UNITED STATES PATENT AND TRADEMARK OFFICE

MN

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/557,250	04/24/2000	Jonathan S. Goldick	MSFT-0174/150793.1	8456

41505 7590 06/05/2007  
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)  
CIRA CENTRE, 12TH FLOOR  
2929 ARCH STREET  
PHILADELPHIA, PA 19104-2891

EXAMINER
----------

PATEL, HARESH N

ART UNIT	PAPER NUMBER
----------	--------------

2154

MAIL DATE	DELIVERY MODE
-----------	---------------

06/05/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

## Office Action Summary

Application No.

09/557,250

Applicant(s)

GOLDICK ET AL.

Examiner

Haresh Patel

Art Unit

2154

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 07 March 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-13, 16-20 and 22-27 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-13, 16-20, 22-27 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

1. Claims 1-13, 16-20, and 22-27 are presented for examination. Claims 14, 15 and 21 are cancelled.

#### ***Response to Arguments***

2. Applicant's arguments with respect to claims 1-13, 16-20, and 22-27 have been considered but are moot in view of the new ground(s) of rejection.

#### ***Double Patenting***

3. Claims 1-13, 16-20, and 22-27 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-59 of Lomet, U.S. Patent No. 5,870,763, as per the previous office action dated 9/7/2006. Regarding the applicant's concern regarding, "registering applications ..., with an ... for communication of application's state dependency information ...among..." for further explanation, the patent claims contain computer system that makes applications recoverable from system crashes with tracking whether the application object i.e., application's state has any flush order dependencies with other objects, usage of resource manager, cache manager to log application object state information, usage of log and non-volatile memory, communicating the recovery information among recovery components for the recovery of the system, handling the application object state information for the system to recover applications from crash, etc., that accomplish the concerned acts. Regarding the concerned limitations, for the concept of handling dependency among applications, in fact, the abstract of the Lomet contains, making applications recoverable from

Art Unit: 2154

system crashes, the application state (i.e., address space) is treated as a single object, ...., applications are recovered by replaying the logged state transition operations, that accomplished the handling of the dependency among applications. Regarding, the well-known concept of using application programming interface, API, first, Lomet clearly states, "The invention has been described in language more or less specific as to structure and method features. It is to be understood, however, that the invention is not limited to the specific features described, since the means herein disclosed comprise exemplary forms of putting the invention into effect. The invention is, therefore, claimed in any of its forms or modifications within the proper scope of the appended claims appropriately interpreted in accordance with the doctrine of equivalents and other applicable judicial doctrines", meaning that the Lomet's claimed method etc., can be implemented using all the means (including all well-known techniques, usage of API, etc.) and even though Lomet claims did not specifically contain API to implement the steps for the recovery of the applications using the components of the recovery system, the Lomet claimed method steps, etc., are patented against all who try to implement the Lomet claims with addition of well-known usage of API. Adding well-known usage of API to Lomet claims does not mean the API itself becomes novel and the claimed subject matter of the Lomet loses the patented subject matter.

One of ordinary skilled in the art at the time of invention very well knows what the API is and the usage of the API. The API is a set of routines used by an application program to direct the performance of procedures by the computer's operating system.

Lomet's recovery system for recovering applications from crash according to the dependency information before the crash is implemented using software and one of ordinary

skilled in the art at the time of invention very well knows that the API can be used for the implementation. Use of API is well-known in the art, for example, Bacha et al., IBM, 6,839,843, col., 5; Chou et al., Intel, 5,832, 283, col., 8; Klein et al., 6,209,000, col., 5; discloses usage of the well-known concept of using API. It would be obvious to one of ordinary skill in the art to include concept of using API with the claimed subject matter of the claims of Lomet. The API would support handling of software modules for the recovery of the applications based on the dependency information. The usage of the software modules would support accomplishing recovery of the system after crash to put the applications back in the pre-crash state.

“A later patent claim is not patentably distinct from an earlier patent claim if the later claim is obvious over, or anticipated by, the earlier claim”.

**In re Longi, 759 F.2d at 896, 225 USPQ at 651** (affirming a holding of obviousness-type double patenting because the claims at issue were obvious over claims in four prior art patents); **In re Bern, 140 F.3d at 1437, 46 USPQ2d at 1233** (Fed. Cir. 1998) (affirming a holding of obviousness-type double patenting where a patent application claim to a genus is anticipated by a patent claim to a species within that genus). " **ELI LILLY AND COMPANY v BARR LABORATORIES, INC.**, United States Court of Appeals for the Federal Circuit, ON PETITION FOR REHEARING EN BANC (DECIDED: May 30, 2001).

4. Claims 1-13, 16-20, and 22-27 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-39 of Lomet, U.S. Patent No. 5,946,698, as per the previous office action dated 9/7/2006. Regarding the applicant's concern regarding, “registering applications ..., “storing ... at least one application's state

Art Unit: 2154

dependency information ..., “ and “communicating said ... information”, “handling dependency information”, the patent claims contain computer system that makes applications recoverable from system crashes with usage of the application object, usage of resource manager, cache manager to log application object state information, usage of log and non-volatile memory, communicating the recovery information among recovery components for the recovery of the system, handling the application object state information for the system to recover applications from crash, etc., that accomplish the concerned acts. Regarding the concerned limitations, for the concept of handling dependency among applications, in fact, the abstract of the Lomet contains, making applications recoverable from system crashes, the application state (i.e., address space) is treated as a single object, ....., applications are recovered by replaying the logged state transition operations, that accomplished the handling of the dependency among applications. Regarding, the well-known concept of using application programming interface, API, first, Lomet clearly states, “The invention has been described in language more or less specific as to structure and method features. It is to be understood, however, that the invention is not limited to the specific features described, since the means herein disclosed comprise exemplary forms of putting the invention into effect. The invention is, therefore, claimed in any of its forms or modifications within the proper scope of the appended claims appropriately interpreted in accordance with the doctrine of equivalents and other applicable judicial doctrines”, meaning that the Lomet’s claimed method etc., can be implemented using all the means (including all well-known techniques, usage of API, etc.) and even though Lomet claims did not specifically contain API to implement the steps for the recovery of the applications using the components of the recovery system, the Lomet claimed method steps, etc., are patented against all who try to implement the

Art Unit: 2154

Lomet claims with addition of well-known usage of API. Adding well-known usage of API to Lomet claims does not mean the API itself becomes novel and the claimed subject matter of the Lomet loses the patented subject matter.

One of ordinary skilled in the art at the time of invention very well knows what the API is and the usage of the API. The API is a set of routines used by an application program to direct the performance of procedures by the computer's operating system.

Lomet's recovery system for recovering applications from crash according to the dependency information before the crash is implemented using software and one of ordinary skilled in the art at the time of invention very well knows that the API can be used for the implementation. Use of API are well-known in the art, for example, Bacha et al., IBM, 6,839,843, col., 5; Chou et al., Intel, 5,832, 283, col., 8, Klein et al., 6,209,000, col., 5; discloses usage of the well-known concept of using API. It would be obvious to one of ordinary skill in the art to include concept of using API with the claimed subject matter of the claims of Lomet. The API would support handling of software modules for the recovery of the applications based on the dependency information. The usage of the software modules would support accomplishing recovery of the system after crash to put the applications back in the pre-crash state.

5. Claims 1-13, 16-20, and 22-27 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-46 of Lomet, U.S. Patent No. 6,067,550, as per the previous office action dated 9/7/2006. Regarding the applicant's concern regarding, "registering applications ..., ""storing ... at least one application's state dependency information ..., "" and "communicating said ... information", "handling dependency

Art Unit: 2154

information”, the patent claims contain computer system that makes applications recoverable from system crashes with handling of the application object i.e., application’s state has any flush order dependencies with other objects, usage of resource manager, cache manager to log application object state information, usage of logging and non-volatile memory, communicating the recovery information among recovery components for the recovery of the system, handling the application object state information for the system to recover applications from crash, etc., that accomplish the concerned acts. Regarding the concerned limitations, for the concept of handling dependency among applications, in fact, the abstract of the Lomet contains, making applications recoverable from system crashes, the application state (i.e., address space) is treated as a single object, ....., applications are recovered by replaying the logged state transition operations, that accomplished the handling of the dependency among applications. Regarding, the well-known concept of using application programming interface, API, first, Lomet clearly states, “The invention has been described in language more or less specific as to structure and method features. It is to be understood, however, that the invention is not limited to the specific features described, since the means herein disclosed comprise exemplary forms of putting the invention into effect. The invention is, therefore, claimed in any of its forms or modifications within the proper scope of the appended claims appropriately interpreted in accordance with the doctrine of equivalents and other applicable judicial doctrines”, meaning that the Lomet’s claimed method etc., can be implemented using all the means (including all well-known techniques, usage of API, etc.) and even though Lomet claims did not specifically contain API to implement the steps for the recovery of the applications using the components of the recovery system, the Lomet claimed method steps, etc., are patented against all who try to implement the



Art Unit: 2154

Lomet claims with addition of well-known usage of API. Adding well-known usage of API to Lomet claims does not mean the API itself becomes novel and the claimed subject matter of the Lomet loses the patented subject matter.

One of ordinary skilled in the art at the time of invention very well knows what the API is and the usage of the API. The API is a set of routines used by an application program to direct the performance of procedures by the computer's operating system.

Lomet's recovery system for recovering applications from crash according to the dependency information before the crash is implemented using software and one of ordinary skilled in the art at the time of invention very well knows that the API can be used for the implementation. Use of API are well-known in the art, for example, Bacha et al., IBM, 6,839,843, col., 5; Chou et al., Intel, 5,832, 283, col., 8, Klein et al., 6,209,000, col., 5; discloses usage of the well-known concept of using API. It would be obvious to one of ordinary skill in the art to include concept of using API with the claimed subject matter of the claims of Lomet. The API would support handling of software modules for the recovery of the applications based on the dependency information. The usage of the software modules would support accomplishing recovery of the system after crash to put the applications back in the pre-crash state.

6. Claims 1, 16 and 22 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-6 of Lomet, U.S. Patent No. 6,151,607, as per the previous office action dated 9/7/2006. Regarding the applicant's concern regarding "registering applications ..., with an ... for communication of application's state dependency

Art Unit: 2154

information ...among...”, the patent claims contain computer system that makes applications recoverable from system crashes with usage of the application object, usage of resource manager, cache manager to log application object state information, usage of log and non-volatile memory, communicating the recovery information among recovery components for the recovery of the system, handling the application object state information for the system to recover applications from crash, etc., that accomplish the concerned acts. Regarding the concerned limitations, for the concept of handling dependency among applications, in fact, the abstract of the Lomet contains, making applications recoverable from system crashes, the application state (i.e., address space) is treated as a single object, ....., applications are recovered by replaying the logged state transition operations, that accomplished the handling of the dependency among applications. Regarding, the well-known concept of using application programming interface, API, first, Lomet clearly states, “The invention has been described in language more or less specific as to structure and method features. It is to be understood, however, that the invention is not limited to the specific features described, since the means herein disclosed comprise exemplary forms of putting the invention into effect. The invention is, therefore, claimed in any of its forms or modifications within the proper scope of the appended claims appropriately interpreted in accordance with the doctrine of equivalents and other applicable judicial doctrines”, meaning that the Lomet’s claimed method etc., can be implemented using all the means (including all well-known techniques, usage of API, etc.) and even though Lomet claims did not specifically contain API to implement the steps for the recovery of the applications using the components of the recovery system, the Lomet claimed method steps, etc., are patented against all who try to implement the Lomet claims with addition

Art Unit: 2154

of well-known usage of API. Adding well-known usage of API to Lomet claims does not mean the API itself becomes novel and the claimed subject matter of the Lomet loses the patented subject matter.

One of ordinary skilled in the art at the time of invention very well knows what the API is and the usage of the API. The API is a set of routines used by an application program to direct the performance of procedures by the computer's operating system.

Lomet's recovery system for recovering applications from crash according to the dependency information before the crash is implemented using software and one of ordinary skilled in the art at the time of invention very well knows that the API can be used for the implementation. Use of API are well-known in the art, for example, Bacha et al., IBM, 6,839,843, col., 5; Chou et al., Intel, 5,832,283, col., 8; Klein et al., 6,209,000, col., 5; discloses usage of the well-known concept of using API. It would be obvious to one of ordinary skill in the art to include concept of using API with the claimed subject matter of the claims of Lomet. 5,946,698. The API would support handling of software modules for the recovery of the applications based on the dependency information. The usage of the software modules would support accomplishing recovery of the system after crash to put the applications back in the pre-crash state.

### ***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter, which the applicant regards as his invention.

Art Unit: 2154

7. Following claims are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The term "common" in claim 1 is a relative term, which renders the claim indefinite.

Claim 3 recites the limitations, "said backup service (entity) includes a determination (step)". These limitations are indefinite for failing to particularly point out and distinctly claim the subject matter. (entity including step). Similar applies to claim 4 that contain "said backup service includes execution". Similar applies to claim 19

Claim 4 recites the limitations, "the freezing of applications", "the determined application freeze order". There is insufficient antecedent basis for this limitation in the claim (Please see MPEP 706.03(d)).

Claim 6 recites the limitations, "the restore operation". There is insufficient antecedent basis for this limitation in the claim (Please see MPEP 706.03(d)).

Claim 7 recites the limitations, "the minimum set of information", "the restore information". There is insufficient antecedent basis for this limitation in the claim (Please see MPEP 706.03(d)). Similar applies to claim 27.

### ***Claim Rejections - 35 USC § 103***

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2154

9. Claims 1-9, 12, 13, 16-20 and 22-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lomet et al. 5,870,763 (Hereinafter Lomet) in view of Bacha et al., IBM, 6,839,843 (Hereinafter Bacha-IBM).

10. As per claim 1, Lomet teaches the following:

a method for utilizing application's state dependency information (e.g., col., 6, lines 41 – 58) to efficiently perform a backup service operation (e.g., col., 7, lines 6 – 26) in a computer system (e.g., col. 5, lines 31 – 46), comprising the acts of:

registering applications (e.g., col., 6, lines 3 – 26) loaded in said computer system (e.g., col. 5, lines 31 – 46) with a software module (e.g., col., 34, lines 21 – 47) for communications of application's state dependency (e.g., col., 6, lines 41 – 58) information among objects (e.g., col., 6, lines 32 – 45), a common software agent (e.g., col., 5, lines 60 – 67), a storage component (e.g., col., 6, lines 41 – 56) utilized by said agent (e.g., col., 5, lines 60 – 67) and a backup service (e.g., col., 5, lines 40 – 51),

storing in said storage component (e.g., col., 6, lines 41 – 56) at least one application's state dependency information (e.g., col., 6, lines 41 – 58) and

communicating said at least one application's state dependency information (e.g., col., 6, lines 41 – 58) from said storage component (e.g., col., 6, lines 41 – 56) to said backup service (e.g., col., 5, lines 40 – 51);

Lomet also disclose the dependency among applications as claimed (e.g., abstract).

However Lomet does not specifically mention about usage of application programming interface (API).

Bacha-IBM discloses the well-known concept of using application programming interface, API (e.g., col., 5).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Lomet with the teachings of Bacha-IBM in order to facilitate using application programming interface (API) because the API would support handling of software modules for the recovery of the applications based on the dependency information. The usage of the software modules would support accomplishing recovery of the system after crash to put the applications back in the pre-crash state.

11. As per claim 2, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

said backup service includes a snapshot service (e.g., col., 11, lines 60 - 66).

12. As per claim 3, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

said backup service includes a determination of an application freeze order (e.g., col.6, lines 41-49).

13. As per claim 4, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

said backup service includes an execution of the freezing of applications in the order reflected by the determined application freeze order (e.g., col. 17, lines 3-42).

Art Unit: 2154

14. As per claim 5, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

loading the software module into said computer system (e.g., col. 19, lines 9-34).

15. As per claim 6, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

said backup service requesting a set of application dependency information from a common software agent for use in connection with the restore operation (e.g., col. 31, lines 32-46).

16. As per claim 7, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

said set of application dependency information is the minimum set of information from said storage component for successfully completing the restore operation (e.g., col. 31, lines 32-46).

17. As per claim 8, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

said agent issuing a request to at least one registered application for information from said set of application dependency information requested by the service (e.g., col. 33, lines 6-42).

18. As per claim 9, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

at least one registered application communicating information to said agent in response to a request by said agent (e.g., col. 12, lines 59-65, figure 5), said information relating to said at least one application's state dependency information (e.g., col. 19, lines 8-14, figure 12).

19. As per claim 12, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said agent stores said at least one application's state dependency information in a tabular format reflective of hierarchical application dependencies in said storage component (e.g., col. 18, lines 4-28, also Van-IBM, abstract).

20. As per claim 13, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

a computer-readable medium having executable instructions for instructing a client computer to perform the acts of the method (e.g., figure 3).

21. As per claim 16, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

in response to a request by a service (e.g., col. 31, lines 32-46) and thereafter delivers said application's state dependency information to said service for further processing by said service (e.g., col., 11, lines 60 - 66).



Art Unit: 2154

22. As per claim 17, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

the service to which said agent delivers said information is a backup service (e.g., col., 7, lines 6 – 26).

23. As per claim 18, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said service includes a snapshot service (e.g., col., 11, lines 60 - 66).

24. As per claim 19, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said service includes a determination of an application freeze order (e.g., col.6, lines 41-49).

25. As per claim 20, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said agent stores said application dependency information in a tabular format reflective of hierarchical application dependencies in a storage component (e.g., col. 18, lines 4-28, also Van-IBM, abstract).

Art Unit: 2154

26. As per claim 22, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

a computer system (e.g., abstract), comprising:

a plurality of applications loaded in said system (e.g., col. 6, lines 2-38), wherein at least one of said applications has at least one external data dependency associated therewith (e.g., col. 19, lines 8-14, figure 12),

a storage component for storing application dependency information (e.g., col. 6, lines 50-60), wherein said dependency information is configured to include information about said at least one external state dependency (e.g., col. 6, lines 50-60, col. 19, lines 8-14, figure 12);

an agent (e.g., col., 5, lines 60 – 67) that functions in said system (e.g., col. 12, lines 43-50) for processing said dependency information (e.g., col., 6, lines 41 – 58), wherein said dependency information includes information about dependencies executing on the system (e.g., col. 6, lines 50-60, col. 19, lines 8-14, figure 12) communicated to the software module from said agent (e.g., col. 12, lines 43-50) and for storing the dependency information in said storage component (e.g., col., 6, lines 41 – 56) and

a service for making requests (e.g., col. 31, lines 32-46) to said agent for a set of dependency information (e.g., col., 6, lines 41 – 58), wherein said agent collects (e.g., (e.g., col., 5, lines 40 – 51), stores (e.g., col., 6, lines 41 – 56) and packages (e.g., col., 6, lines 32 – 45) said dependency information (e.g., col., 6, lines 41 – 58) in response to a request by said service (e.g., col. 19, lines 8-14, figure 12) and delivers said set of dependency information (e.g., col., 6, lines 41 – 58) to said service for further processing by said service (e.g., col. 31, lines 32-46).

Art Unit: 2154

27. As per claim 23, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

the service to which said agent delivers said dependency information is a backup service (e.g., col., 7, lines 6 – 26).

28. As per claim 24, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said service includes a snapshot service (e.g., col., 11, lines 60 - 66).

29. As per claim 25, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said service includes a determination of an application freeze order (e.g., col.6, lines 41-49).

30. As per claim 26, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said agent stores said at least one application's state dependency information in a tabular format reflective of hierarchical application dependencies in a storage component (e.g., col. 18, lines 4-28, also Van-IBM, abstract).

31. As per claim 27, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

Art Unit: 2154

said set of application dependency information is the minimum set of information from said storage component for successfully completing the service (e.g., col. 31, lines 32-46).

32. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lomet and Bacha-IBM in view of “Official Notice”.

33. As per claim 10, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. However, Lomet and Bacha-IBM do not specifically mention about unregistering an application. “Official Notice” is taken that both the concept and advantages of unregistering an application is well known and expected in the art.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include unregistering an application with the teachings of Lomet and Bacha-IBM in order to facilitate unregistering an application because the unregistering would support deregistration of the application. The deregistered application would no longer be used by the system until the application is registered again.

34. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lomet and Bacha-IBM in view of Lewis 6,513,019.

35. As per claim 11, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. However, Lomet and Bacha-IBM do not specifically mention about using XML protocol.

Lewis discloses usage of a communications format comprising XML (e.g., paragraph 27).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Lomet and Bacha-IBM with the teachings of Lewis in

Art Unit: 2154

order to facilitate usage of XML protocol because the XML protocol would provide a web standard common middleware layer in a communication stack at the API level between objects.

36. Claims 1-9, 12, 13, 16-20 and 22-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lomet et al. 5,946,698 (Hereinafter Lomet) in view of Bacha-IBM.

37. As per claim 1, Lomet teaches the following:

a method for utilizing application's state dependency information (e.g., col., 6, lines 41 – 58) to efficiently perform a backup service operation (e.g., col., 7, lines 6 – 26) in a computer system (e.g., col. 5, lines 31 – 46), comprising the acts of:

registering applications (e.g., col., 6, lines 3 – 26) loaded in said computer system (e.g., col. 5, lines 31 – 46) with a software module (e.g., col., 34, lines 21 – 47) for communications of application's state dependency (e.g., col., 6, lines 41 – 58) information among objects (e.g., col., 6, lines 32 – 45), a common software agent (e.g., col., 5, lines 60 – 67), a storage component (e.g., col., 6, lines 41 – 56) utilized by said agent (e.g., col., 5, lines 60 – 67) and a backup service (e.g., col., 5, lines 40 – 51),

storing in said storage component (e.g., col., 6, lines 41 – 56) at least one application's state dependency information (e.g., col., 6, lines 41 – 58) and

communicating said at least one application's state dependency information (e.g., col., 6, lines 41 – 58) from said storage component (e.g., col., 6, lines 41 – 56) to said backup service (e.g., col., 5, lines 40 – 51);

Lomet also disclose the dependency among applications as claimed (e.g., abstract).

However Lomet does not specifically mention about usage of application programming interface (API).

Bacha-IBM discloses the well-known concept of using application programming interface, API (e.g., col., 5).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Lomet with the teachings of Bacha-IBM in order to facilitate using application programming interface (API) because the API would support handling of software modules for the recovery of the applications based on the dependency information. The usage of the software modules would support accomplishing recovery of the system after crash to put the applications back in the pre-crash state.

38. As per claim 2, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said backup service includes a snapshot service (e.g., col., 11, lines 60 - 66).

39. As per claim 3, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said backup service includes a determination of an application freeze order (e.g., col.6, lines 41-49).

40. As per claim 4, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

Art Unit: 2154

said backup service includes an execution of the freezing of applications in the order reflected by the determined application freeze order (e.g., col. 17, lines 3-42).

41. As per claim 5, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

loading the software module into said computer system (e.g., col. 19, lines 9-34).

42. As per claim 6, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

said backup service requesting a set of application dependency information from a common software agent for use in connection with the restore operation (e.g., col. 31, lines 32-46).

43. As per claim 7, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

said set of application dependency information is the minimum set of information from said storage component for successfully completing the restore operation (e.g., col. 31, lines 32-46).

44. As per claim 8, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

said agent issuing a request to at least one registered application for information from said set of application dependency information requested by the service (e.g., col. 33, lines 6-42).

45. As per claim 9, Lomet and Bacha-IBM disclose the claimed limitations as rejected above.

Lomet also teaches the following:

at least one registered application communicating information to said agent in response to a request by said agent (e.g., col. 12, lines 59-65, figure 5), said information relating to said at least one application's state dependency information (e.g., col. 19, lines 8-14, figure 12).

46. As per claim 12, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said agent stores said at least one application's state dependency information in a tabular format reflective of hierarchical application dependencies in said storage component (e.g., col. 18, lines 4-28, also Van-IBM, abstract).

47. As per claim 13, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

a computer-readable medium having executable instructions for instructing a client computer to perform the acts of the method (e.g., figure 3).

48. As per claim 16, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:



Art Unit: 2154

in response to a request by a service (e.g., col. 31, lines 32-46) and thereafter delivers said application's state dependency information to said service for further processing by said service (e.g., col., 11, lines 60 - 66).

49. As per claim 17, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

the service to which said agent delivers said information is a backup service (e.g., col., 7, lines 6 – 26).

50. As per claim 18, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said service includes a snapshot service (e.g., col., 11, lines 60 - 66).

51. As per claim 19, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said service includes a determination of an application freeze order (e.g., col.6, lines 41-49).

52. As per claim 20, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

Art Unit: 2154

said agent stores said application dependency information in a tabular format reflective of hierarchical application dependencies in a storage component (e.g., col. 18, lines 4-28, also Van-IBM, abstract).

53. As per claim 22, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

a computer system (e.g., abstract), comprising:

a plurality of applications loaded in said system (e.g., col. 6, lines 2-38), wherein at least one of said applications has at least one external data dependency associated therewith (e.g., col. 19, lines 8-14, figure 12),

a storage component for storing application dependency information (e.g., col. 6, lines 50-60), wherein said dependency information is configured to include information about said at least one external state dependency (e.g., col. 6, lines 50-60, col. 19, lines 8-14, figure 12);

an agent (e.g., col., 5, lines 60 – 67) that functions in said system (e.g., col. 12, lines 43-50) for processing said dependency information (e.g., col., 6, lines 41 – 58), wherein said dependency information includes information about dependencies executing on the system (e.g., col. 6, lines 50-60, col. 19, lines 8-14, figure 12) communicated to the software module from said agent (e.g., col. 12, lines 43-50) and for storing the dependency information in said storage component (e.g., col., 6, lines 41 – 56) and

a service for making requests (e.g., col. 31, lines 32-46) to said agent for a set of dependency information (e.g., col., 6, lines 41 – 58), wherein said agent collects (e.g., (e.g., col., 5, lines 40 – 51), stores (e.g., col., 6, lines 41 – 56) and packages (e.g., col., 6, lines 32 – 45) said

Art Unit: 2154

dependency information (e.g., col., 6, lines 41 – 58) in response to a request by said service (e.g., col. 19, lines 8-14, figure 12) and delivers said set of dependency information (e.g., col., 6, lines 41 – 58) to said service for further processing by said service (e.g., col. 31, lines 32-46).

54. As per claim 23, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

the service to which said agent delivers said dependency information is a backup service (e.g., col., 7, lines 6 – 26).

55. As per claim 24, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said service includes a snapshot service (e.g., col., 11, lines 60 - 66).

56. As per claim 25, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said service includes a determination of an application freeze order (e.g., col.6, lines 41-49).

57. As per claim 26, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said agent stores said at least one application's state dependency information in a tabular format reflective of hierarchical application dependencies in a storage component (e.g., col. 18, lines 4-28, also Van-IBM, abstract).

58. As per claim 27, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. Lomet also teaches the following:

said set of application dependency information is the minimum set of information from said storage component for successfully completing the service (e.g., col. 31, lines 32-46).

59. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lomet and Bacha-IBM in view of "Official Notice".

60. As per claim 10, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. However, Lomet and Bacha-IBM do not specifically mention about unregistering an application. "Official Notice" is taken that both the concept and advantages of unregistering an application is well known and expected in the art.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include unregistering an application with the teachings of Lomet and Bacha-IBM in order to facilitate unregistering an application because the unregistering would support deregistration of the application. The deregistered application would no longer be used by the system until the application is registered again.

Art Unit: 2154

61. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lomet and Bacha-IBM in view of Lewis.

62. As per claim 11, Lomet and Bacha-IBM disclose the claimed limitations as rejected above. However, Lomet and Bacha-IBM do not specifically mention about using XML protocol.

Lewis discloses usage of a communications format comprising XML (e.g., paragraph 27).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Lomet and Bacha-IBM with the teachings of Lewis in order to facilitate usage of XML protocol because the XML protocol would provide a web standard common middleware layer in a communication stack at the API level between objects.

### *Conclusion*

Larsson et al., 5,530,800 (Hereinafter Larsson) also discloses the concept of handling dependency among applications (e.g., usage of dependency information among sub-systems, col., 4, line 58 – col., 5, line 26).

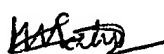
Examiner has cited particular columns and line numbers and/or paragraphs and/or sections and/or page numbers in the reference(s) as applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in entirety, as potentially teaching, all or part of the claimed invention, as well as the context of the passage, as taught by the prior art or disclosed by the Examiner.

Art Unit: 2154

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Haresh Patel whose telephone number is (571) 272-3973. The examiner can normally be reached on Monday, Tuesday, Thursday and Friday from 10:00 am to 8:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nathan Flynn can be reached on (571) 272-1915. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

 Haresh Patel

Haresh Patel

May 27, 2007